ASTEEC Conference Proceeding: Computer Science
3rd International Conference on Information Science and Technology Innovation (ICoSTEC)
July 27, 2024, Yogyakarta, Indonesia

Comparison of CNN Method Architectures in Arabic Sign Language Image Classification

Ade Umar Ramadhan [1] , Nida Muhliya Barkah [2] , Shofwatul 'Uyun [3]

[1,2,3] *Master of Informatics, Faculty of Science and Technology, UIN Sunan Kalijaga, Yogyakarta, Indonesia*
[1] au.ramadhan24@gmail.com
[2] nidamuhliya35@gmail.com
[3] shofwatul.uyun@ uin-suka.ac.id

***Abstract*— Sign language is an essential communication tool for people with disabilities, especially for deaf and speech-impaired people. Sign language allows people with disabilities to interact and participate actively in social and educational settings. This research compared several CNN method architectures, such as LeNet-5, AlexNet, and VGG-16, in Arabic Sign Language image classification to find the best architecture with the highest accuracy value. The dataset used in this research is the Arabic Alphabets Sign Language Dataset (ArASL), which consists of 47,876 images and is divided into 28 classes. This research's training and testing process uses K-Fold cross-validation with a K-fold value = 5. The testing results are then evaluated using the Confusion Matrix to calculate and obtain the best accuracy value. The results of the research show that the average accuracy value obtained from each fold for the LeNet-5 architecture reaches a value of 97.38%, for the AlexNet architecture, it reaches an accuracy value of 97.96%, and for the VGG-16 architecture, it reaches an accuracy value of 98.17%. Based on the research results, it can be concluded that using VGG-16 architecture shows the best performance and is the most optimal choice in classifying Arabic sign language images on the ArASL dataset compared to LeNet-5 and AlexNet.**

***Keywords*— Sign Language, CNN, ArASL, LeNet-5, Alexnet, VGG-16**

I. Introduction

Sign language uses hand movements and can be seen by the eyes, which is used as an alternative by the disabled or handicapped community, especially for deaf and speech-impaired people [1]. People with disabilities use sign language as a tool to communicate and interact in social life [2]. According to the *World* Fact Sheet *Health Organization* (WHO), around 5% of the world's population experiences hearing loss [3], which seems small but shows that there are more than 460 million people worldwide, 34 million of whom are children. This number is expected to increase by 2050 to 900 million people who will experience hearing loss, and around 1.1 billion young people are at risk of experiencing hearing loss [4]; the cost of untreated hearing loss reaches a cost of 750 billion US dollars [5]. Today, sign language and automatic translation tools are used by approximately 70 million people worldwide, which has a significant impact on the way they communicate with each other [6].

Arabic script is a standard for writing Arabic, generally known as the Arabic alphabet [7]. The Arabic alphabet is used by residents of Arab countries, who make up around 14% of the world's population or around 1 billion people [8]. The Arabic alphabet is not only used by Arab countries. However, it is also widely used in Asia and Africa, around a quarter of the world's population, influencing most of the world's dialects and languages [9]. *Arabic Sign Language* (ArSL) is a sign language used by deaf and speech-impaired people in Arab countries to overcome communication problems using Arabic and enable their involvement in the world of education [10]. ArSL is divided into two types: ArSL and ArASL [11]. ArSL is an Arabic sign language that expresses one word with specific movements, while ArASL (*Arabic Alphabet Sign Language*) is a sign language that spells each letter in words [12]. Several problems and challenges in using Arabic Sign Language arise, so an approach is needed to overcome the complexity of variations in Arabic Sign Language movements.

The convolutional Neural Network (CNN) approach can solve the problems and challenges of Arabic Sign Language. CNN is a type of neural *network* used in *deep learning* [13]. CNN is a type of *neural network* with the main advantages in processing image data [14]. CNN imitates the workings of human brain neural networks and uses kernels to extract input image data features using convolution operations [15]. The layers in CNN consist of *the Convolution Layer, Pooling Layer,* and *Fully Connected Layer* [16].

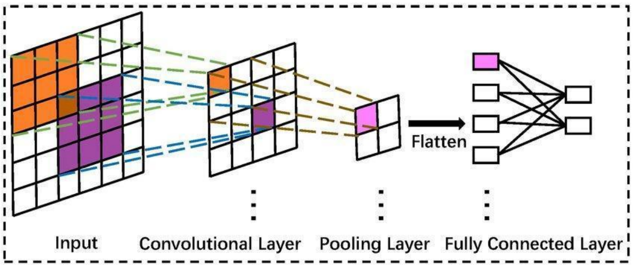

Fig. 1 CNN Composing Layer [17]

The convolution layer is the core layer of CNN, which is responsible for calculating the output of neurons connected to local regions in the input image data. The pooling layer is a layer that reduces the dimensions of the feature map to speed up the computing process and overcome overfitting problems. Flatten reshapes features into vectors for input from a fully connected layer. The following fully connected layer will calculate classification class scores like neural networks in general [18]. CNN has several frequently used architectures, such as LeNet-5, AlexNet, and VGG-16. This architecture has its advantages that can be used to solve the problem of the complexity of sign language movements.

Several previous studies have been carried out using the CNN method to solve the challenges and problems faced in Arabic Sign Language. Research conducted by Alawwad [19] to identify and recognize ArASL used the Faster R-CNN method utilizing the VGG and ResNet-18 models. The results show that the proposed approach produces 93% accuracy and confirms the reliability of the proposed model. Research was also carried out by Ismail [20] to overcome the problem of recognizing Arabic sign language movements in video data using two types of RNN: Long short-term memory (LSTM) and gated recurrent unit (GRU). The experimental results show an accuracy value of 100% on the dataset used. Alyami [21] researched to provide an ArASL recognition model that is light and fast and can be implemented in *real-time applications*. The proposed model is evaluated on the LSA64 dataset and obtains 98.25% and 91.09% accuracy. Research conducted by AbdElghfar [22] proposed a new model for CNN Al-Qur'an sign language recognition, which is aimed at recognizing Arabic sign language movements by recognizing hand movements that refer to the letters of the Koran. Experimental results show that the proposed model is better than other existing models. Research was also conducted by Kamruzzaman [23] using CNN to recognize signs and hand movements used in Arabic sign language. The research results provide 90% accuracy for recognizing Arabic sign language, so the resulting system has high reliability. Several studies were also conducted in Indonesian Sign language. This was done by Kersen [24] to avoid SIBI sign language translation errors using the CNN method, which showed an accuracy value of 52%. Research conducted by Sholawati [25] to develop an SIBI alphabet recognition application using CNN. Application testing results show accuracy, recall, specificity and sensitivity values of 80.76%. Research was also conducted by Thira [26] to classify the SIBI alphabet, which is divided into 24 classes, by comparing three CNN architectures, MobileNetV2, MobileNetV3Small, and MobileNetV3Large. The research results show that MobileNetV3Small is the best model, achieving an accuracy value of 98.81%.

Based on previous research, this research compared several CNN architectures, such as LeNet-5, AlexNet, and VGG-16, to perform ArASL image classification and find the best architecture with the highest accuracy value. It is hoped that the results of this research can help future research related to the development, introduction and classification of Arabic Sign Language.

II. RESEARCH METHODS

This research was conducted to evaluate the results of Arabic sign language image classification using several CNN architectures, such as LeNet-5, AlexNet, and VGG-16, to determine the best method characterized by the highest accuracy value. The system block diagram showing the research flow can be seen in Figure 2.

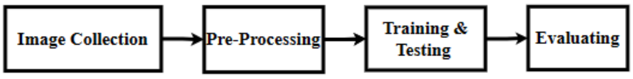

Fig. 2 System Block Diagram

The research flow consists of 4 main stages: Image Collection, Preprocessing, Training and Testing and Evaluating.

A. Image Collection

The dataset used in this research is the Arabic Alphabets Sign Language Dataset (ArASL), which was produced by Latif [27] in previous research. ArASL consists of 47,876**.** The Arabic alphabet sign language image was taken using an iPhone 6s smart camera and divided into 28 classes. The images in the dataset measure 64 x 64 pixels in JPG format. ArASL can be accessed and downloaded for free on the official Mendeley Data website. The dataset image for each class can be seen in Figure 3.

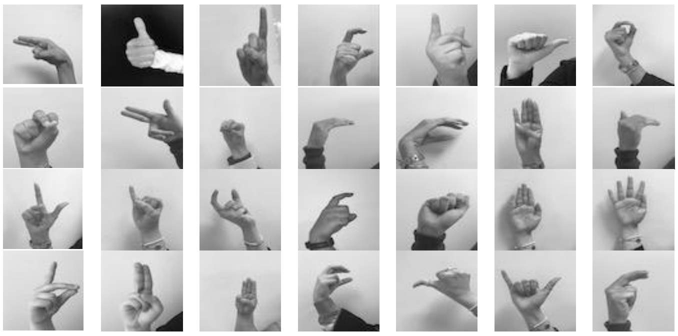

Fig. 3 Images of Arabic Sign Language

B. Preprocessing

Preprocessing is carried out to optimize image quality and simplify and improve the system's ability to carry out classification. Preprocessing in this research was carried out in several stages, such as Label Encoding, Corrupt Image Repair and Data Normalization.

1. Encoding Labels

Label Encoding is used to define classes by identifying the parent folder of each image. This process helps provide a unique numerical representation for each class in the dataset.

2. Corrupt Image Repair

Before resizing the image, repairs are made to the image that may be damaged. Damaged images are converted into grayscale images to ensure data integrity.

3. Data Resize and Normalization

The image data is converted into a numpy array and normalized. The image resizing process is carried out to suit the requirements of each architecture to be used. The image is changed to 32×32 pixels for the LeNet-5 model, while for the AlexNet and VGG-16 models, the image size is changed to 224×224 pixels. This normalization process helps in preparing consistent input data for model training.

After the preprocessing process, the dataset is ready for the model training and testing process.

C. *Training & Testing* Models

training and testing process carried out on the model consists of several stages such as Modeling, Data Augmentation, Compile Model, Model Fit and Validation. The CNN model's modelling stage or architectural design is carried out using the Keras interface provided by the TensorFlow library in the Python programming language. The CNN model was built to classify Arabic alphabet sign language images. The data augmentation stage uses ImageDataGenerator with several parameters such as rotation range, zoom range, width shift range and height shift range, which aims to avoid overfitting the model.

The compiling model stage involves determining key configurations that will influence how the model is trained and evaluated, such as the loss function, optimizer, and metrics. The loss function measures the extent to which the model built can predict the appropriate output and correct class labels. Losses The function used is sparse categorical cross-entropy. Optimizer is an algorithm for adjusting model weights based on loss function values. The optimizer used is Adam. Metrics are used to evaluate model performance during and after the training process. The metric used is accuracy to measure how accurately the model can predict classification classes.

The Model Fit stage measures how well the system training model can generalize data in a way similar to the training data. Some parameters are image data in the form of numpy arrays, classification classes, batch seize, epochs and verbose. The Validation stage uses K-Fold Cross Validation with a K-Fold value = 5. Each training stage, such as Modeling, Data Augmentation, Compile Model, and Model Fit, will always be carried out at each fold to increase the accuracy of model evaluation.

The training and testing model process in this research can be seen in Figure 4.

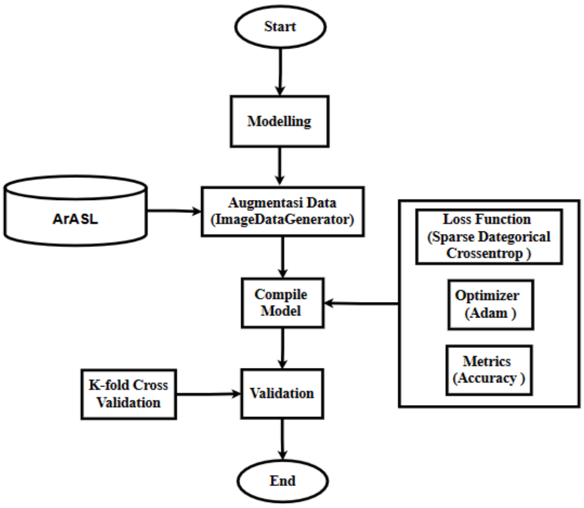

Fig. 4 *Training* & *Testing Process Flow*

D. *Evaluating*

The evaluation process uses a confusion matrix, which calculates the average accuracy for each k-fold to determine the model's accuracy in classifying sign language images according to the classes contained in the dataset. The accuracy value obtained will be a reference in determining the ranking of the best models. The formula for the accuracy value can be seen in equation (1).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (1)$$

III. Results and Discussion

A. Data *Preparation*

This research compared the accuracy of three CNN architecture models, namely LeNet-5, AlexNet, and VGG16. The distribution of data in each class can be seen in Figure 6.

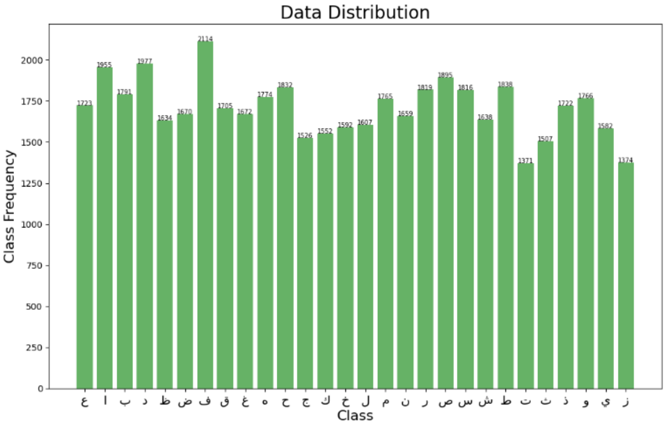

Fig. 6 Data Distribution for each Class in the ArASL Dataset

B. *Train and Test Data*

The training and test stages include model design, data augmentation, model compilation, and model fit. This stage is carried out synchronously with Kfold using *looping*. At the end of each fold, an accuracy evaluation is carried out from the first to the fifth epoch to see the evaluation score for that fold. The accuracy results for each epoch and the evaluation scores for each fold from the first to the fifth fold can be seen in TABLE I to TABLE V.

TABLE 1 shows that the Evaluate Score from epoch 1 to 5 in the First Fold is already above 95% for all three models. The AlexNet method has the highest Evaluate Score of 96.56%, followed by the other two methods, VGG16, which has an accuracy of 95.54% and LeNet-5 at 95.37%.

TABLE I
ACCURACY OF EACH EPOCH ON THE FIRST FOLD

Epoch	LeNet-5	AlexNet	VGG16
1	0.4154	0.5233	0.6881
2	0.8750	0.9164	0.9413
3	0.9350	0.9486	0.9672
4	0.9506	0.9622	0.9804
5	0.9610	0.9670	0.9841
Evaluate Score	**95.37%**	**96.56%**	**95.54%**

TABLE 2 shows that the Evaluate Score from epoch 1 to 5 in the Second Fold of the VGG16 Method has the highest Evaluate Score of 97.94%, followed by the other two methods, namely AlexNet, which has an accuracy of 97.43% and LeNet-5 of 96.45%.

TABLE II
ACCURACY OF EACH EPOCH ON THE SECOND FOLD

Epoch	LeNet-5	AlexNet	VGG16
1	0.9621	0.9720	0.9782
2	0.9686	0.9750	0.9888
3	0.9741	0.9769	0.9925
4	0.9753	0.9773	0.9896
5	0.9803	0.9830	0.9944
Evaluate Score	**96.45%**	**97.43%**	**97.94%**

TABLE 3 shows that the Evaluate Score from epoch 1 to 5 in the Third Fold of the VGG16 Method has the highest Evaluate Score of 98.80%, followed by the other two methods, AlexNet, which has an accuracy of 98.61% and LeNet-5 of 98.25%.

TABLE III
ACCURACY OF EACH EPOCH ON THE THIRD FOLD

Epoch	LeNet-5	AlexNet	VGG16
1	0.9746	0.9826	0.9877
2	0.9812	0.9836	0.9936
3	0.9808	0.9856	0.9953
4	0.9841	0.9844	0.9947
5	0.9837	0.9881	0.9934
Evaluate Score	**98.25%**	**98.61%**	**98.80%**

TABLE 4 shows that the Evaluate Score from epoch 1 to 5 in the Fourth Fold of the VGG16 Method has the highest Evaluate Score of 98.84%, followed by the other two methods, AlexNet, which has an accuracy of 98.42% and LeNet-5 of 97.86%.

TABLE IV
ACCURACY OF EACH EPOCH ON THE FOURTH FOLD

Epoch	LeNet-5	AlexNet	VGG16
1	0.9833	0.9855	0.9921
2	0.9856	0.9872	0.9963
3	0.9861	0.9903	0.9977
4	0.9880	0.9887	0.9954
5	0.9875	0.9907	0.9959
Evaluate Score	**97.86%**	**98.42%**	**98.84%**

TABLE 5 shows that the Evaluate Score from epoch 1 to 5 in the Fifth Fold of the VGG16 Method has the highest Evaluate Score of 99.74%, followed by the other two methods, namely LeNet-5, which has an accuracy of 98.98% and AlexNet of 98.79%.

TABLE V
ACCURACY OF EACH EPOCH ON THE FIFTH FOLD

Epoch	LeNet-5	AlexNet	VGG16
1	0.9853	0.9870	0.9939
2	0.9890	0.9919	0.9957
3	0.9878	0.9903	0.9949
4	0.9898	0.9897	0.9969
5	0.9907	0.9914	0.9975
Evaluate Score	**98.98%**	**98.79%**	**99.74%**

The accuracy obtained from each fold is then averaged to obtain the final accuracy for each model. The LeNet-5 model has a final accuracy of 97.38%, the AlexNet model has a final accuracy of 97.96%, and the VGG-16 has a final accuracy of 98.17%. More complete data for the final accuracy of each model can be seen in TABLE VI.

TABLE VI
FINAL ACCURACY OF EACH MODEL

Epoch	LeNet-5	AlexNet	VGG16
1	95.37%	96.56%	95.54%
2	96.45%	97.43%	97.94%
3	98.25%	98.61%	98.80%
4	97.86%	98.42%	98.84%
5	98.98%	98.79%	99.74%
Average Score	**97.38%**	**97.96%**	**98.17%**

C. *Evaluate Data*

The input and output variables are evaluated against the prediction results that have previously been trained and tested. The evaluation aims to see the match between the original data and the prediction results for each class. In the case of classification, evaluate this using a confusion matrix. The higher the evaluation results on the confusion matrix, the better the model performs classification.

The confusion matrix in the LeNet-5 model has a reasonably high match between actual and predicted data. Evaluation results of the LeNet-5 model can be seen in Figure 7.

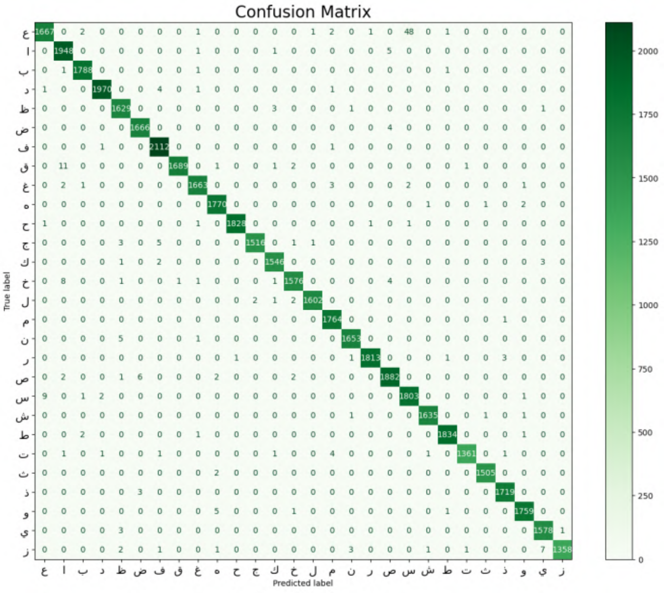

Fig. 7 Confusion Matrix of the LeNet-5 Model

Like the previous model, the confusion matrix in the AlexNet model has a reasonably high match between actual and predicted data. Evaluation results of the AlexNet model can be seen in Figure 8.

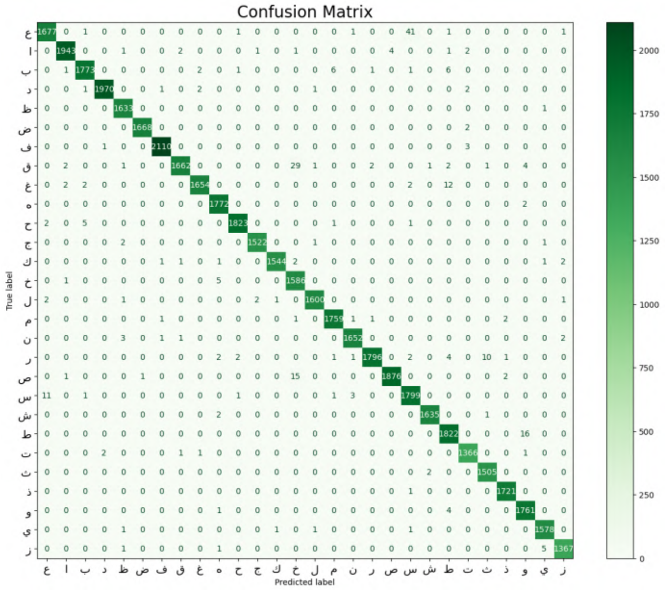

Fig. 8 Confusion Matrix of the AlexNet Model

Finally, the confusion matrix of the VGG16 model has the highest match between actual and predicted data than the previous two models. Evaluation results of the VGG-16 model can be seen in Figure 9.

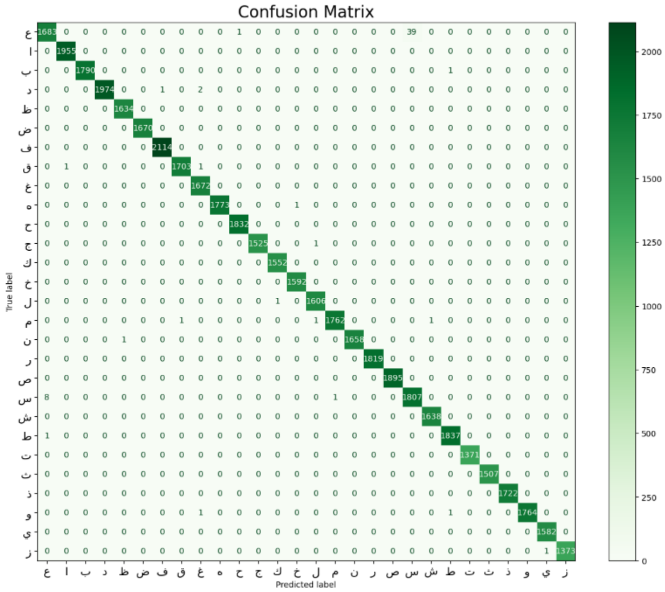

Fig. 7 Confusion Matrix of the VGG-16 Model

Based on Figure 7, Figure 8, and Figure 9, it can be seen that the confusion matrix of each model has good evaluation results because each class has a high score between the actual data and the predicted data.

IV. Conclusions

The results of this research resulted in the conclusion that the models from the CNN architecture, namely LeNet-5, AlexNet, and VGG-16, have perfect accuracy in classifying *Arabic Alphabet Sign Language* (ArASL) image data totalling 47,876 data, which is divided into 28 classes, namely above 95 %. The VGG-16 architecture has the best accuracy among the other architectures, namely 98.17%. In contrast, the accuracy of the other two architectures only has a very slight difference with the previous architecture, namely LeNet-5 at 97.38%, while AlexNet is 97.96%. The accuracy results are also supported by the evaluation of the confusion matrix, which shows that each model class also shows a high match between the actual data and the predicted data.

Acknowledgement

The authors would like to thank the following party for their contributions to this work, UIN Sunan Kalijaga Yogyakarta for their financial support. The authors also thank all the participants for their time and cooperation.

References

[1] Z. Fadhilah and NL Marpaung, "Introduction to the SIBI Alphabet Using Convolutional Neural Network as a Learning Media for the General Public," *J. Inform. J. Developer. IT* , vol. 8, no. 2, pp. 162–168, 2023, doi: 10.30591/jpit.v8i2.5221.

[2] A. Dwi, B. Rizky, and MA Faqihuddin, "Identification of the Indonesian Sign Language Alphabet Using Convolutional LSTM," pp. 183–190, 2023.

[3] "Deafness and Hearing Loss," *World Health Organization*, www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss (accessed Jan. 04, 2024).

[4] M. Zakariah, YA Alotaibi, D. Koundal, Y. Guo, and M. Mamun Elahi, "Sign Language Recognition for Arabic Alphabets Using Transfer Learning Technique," *Comput. Intel. Neurosci.* , vol. 2022, 2022, doi: 10.1155/2022/4567989.

[5] M. Horvat, R.V. Croce, C. Pesce, and A. Fallaize, "Deafness and hearing loss," *Dev. Adapt. Phys. Educ.* , no. June, pp. 217–233, 2019, doi: 10.4324/9780203704035-15.

[6] M. Mustafa, "Retraction Note to: A study on Arabic sign language recognition for differently abled using advanced machine learning classifiers," *J. Ambient Intell. Humaniz. Comput.* , vol. 14, p. 1, 2022, doi: 10.1007/s12652-022-04142-y.

[7] K. Mahdi Hama Rawf, A. Ali Mohammed, A. Othman Abdulrahman, P. Ahmed Abdalla, and KJ Ghafor, "A Comparative Technique Using 2D CNN and Transfer Learning to Detect and Classify Arabic-Script-Based Sign Language, " *Acta Inform. Malaysia* , vol. 7, no. 1, pp. 8–14, 2023, doi: 10.26480/aim.01.2023.08.14.

[8] Z. Alsaadi, E. Alshamani, M. Alrehaili, A. A. D. Alrashdi, S. Albelwi, and A. O. Elfaki, "A Real Time Arabic Sign Language Alphabets (ArSLA) Recognition Model Using Deep Learning Architecture," *Computers* , vol. 11, no. 5, 2022, doi: 10.3390/computers11050078.

[9] M. Mirdehghan Farashah, "Persian, Urdu, and Pashto: A comparative orthographic analysis," *Writ. Syst. Res.* , vol. 2, pp. 9–23, 2010, doi: 10.1093/wsr/wsq005.

[10] N. El-Bendary, HM Zawbaa, MS Daoud, AE Hassanien, and K. Nakamatsu, "ArSLAT: Arabic Sign Language Alphabets Translator," *2010 Int. Conf. Comput. Inf. Syst. Ind. Manag. Appl. CISIM 2010* , no. May 2014, pp. 590–595, 2010, doi: 10.1109/CISIM.2010.5643519.

[11] NM Alharthi and SM Alzahrani, "Vision Transformers and Transfer Learning Approaches for Arabic Sign Language Recognition," *Appl. Sci.* , vol. 13, no. 21, p. 11625, 2023, doi: 10.3390/app132111625.

[12] RM Mohammed and SM Kadhem, "A Review on Arabic Sign Language Translator Systems," *J. Phys. Conf. Ser.* , vol. 1818, no. 1, 2021, doi: 10.1088/1742-6596/1818/1/012033.

[13] IF Alam, MI Sarita, and AMS Sajiah, "Implementation of Deep Learning with the Convolutional Neural Network Method for Real Time Object Identification Based on Android," *semanTIK* , vol. 5, no. 2, pp. 12–26, 2020.

[14] O. Saputra, DI Mulyana, and MB Yel, "Implementation of the Convolutional Neural Network (CNN) Algorithm for the Classification of Traditional Weapons in Central Java Using the Transfer Learning Method," *J. SISKOM-KB (Computing Systems and Artificial Intelligence)* , vol. 5, no. 2, pp. 45–52, 2022, doi: 10.47970/siskom-kb.v5i2.282.

[15] D. Iskandar Mulyana and Wartono, "Optimization of Image Classification Using the Convolutional Neural Network (CNN) Algorithm for Cirebon Batik Image Indonesian," *Int. J. Sci. Eng. Appl. Sci.* , no. 7, p. 12, 2021, [Online]. Available: www.ijseas.com.

[16] MA Hanin, R. Patmasari, RYN Fuâ, and others, "Skin Disease Classification System Using Convolutional Neural Network (cnn)," *eProceedings Eng.* , vol. 8, no. 1, pp. 273–281, 2021.

[17] J. Lu *et al.* , "Automated Strabismus Detection for Telemedicine Applications," no. December, 2018, [Online]. Available: http://arxiv.org/abs/1809.02940.

[18] DS Wita and DY Liliana, "Identity Classification with Palm Images Using Convolutional Neural Network (CNN)," *J. Rekayasa Teknol. Inf.* , vol. 6, no. 1, p. 1, 2022, doi: 10.30872/jurti.v6i1.7100.

[19] RA Alawwad, O. Bchir, and MM Ben Ismail, "Arabic Sign Language Recognition using Faster R-CNN," *Int. J. Adv. Comput. Sci. Appl.* , vol. 12, no. 3, pp. 692–700, 2021, doi: 10.14569/IJACSA.2021.0120380.

[20] MH Ismail, SA Dawwd, and FH Ali, "Dynamic hand gesture recognition of Arabic sign language by using deep convolutional neural networks," *Indonesia. J. Electr. Eng. Comput. Sci.* , vol. 25, no. 2, pp. 952–962, 2022, doi: 10.11591/ijeecs.v25.i2.pp952-962.

[21] S. Alyami, H. Luqman, and M. Hammoudeh, "Isolated Arabic Sign Language Recognition Using A Transformer-based Model and Landmark Keypoints," *ACM Trans. Asian Low-Resource Lang. Inf. Process.* , 2023, doi: 10.1145/3584984.

[22] HA Abdelghfar *et al.* , "A Model for Qur'anic Sign Language Recognition Based on Deep Learning Algorithms," *J. Sensors* , vol. 2023, 2023, doi: 10.1155/2023/9926245.

[23] MM Kamruzzaman, "Arabic Sign Language Recognition and Generating Arabic Speech Using Convolutional Neural Network," *Wirel. Commun. Mob. Comput.* , vol. 2020, 2020, doi: 10.1155/2020/3685614.

[24] K. Kersen and W. Widhiarso, "Application of Convolutional Neural Network Methods in Sign Language Classification," *MDP Student Conf.* , vol. 2, no. 1, pp. 244–249, 2023, doi: 10.35957/mdp-sc.v2i1.4221.

[25] M. Sholawati, K. Auliasari, and F. Ariwibisono, "Development of the Sibi Alphabet Sign Language Recognition Application Using the Convolutional Neural Network (Cnn) Method," *JATI (Journal of Mhs. Tech. Inform.* , vol. 6, no. 1, pp. 134–144, 2022, doi: 10.36040/jati.v6i1.4507.

[26] IJ Thira, D. Riana, AN Ilhami, B. Rizky, and S. Dwinanda, "Introduction to the Indonesian Sign Language System (SIBI) Alphabet Using Convolutional Neural Network," 2019.

[27] G. Latif, N. Mohammad, J. Alghazo, R. AlKhalaf, and R. AlKhalaf, "ArASL: Arabic Alphabets Sign Language Dataset," *Data Br.* , vol. 23, p. 103777, 2019, doi: 10.1016/j.dib.2019.103777.